REMARKS

The Examiner has rejected pending Claims 1 and 11 under 35 USC § 103 as unpatentable over the combined teachings of the Zolnowsky patent, the Custer reference and the admitted prior art (AAPA) regarding the AIX technology. In addition, the Examiner has rejected Claims 1, 11, 12 and 14 under 35 USC § 103 as unpatentable over the teachings of the Boland patent in combination with the teachings of the Custer reference and the AAPA. Claims 2, 4-10, 14-16 and 18 stand rejected as unpatentable over Boland in view of Custer and the AAPA and further in view of the teachings of the Cameron patent. Finally, Claims 3 and 17 have been rejected as unpatentable over Boland in view of Custer, the AAPA, Cameron, and Ripps.

Once again, the Examiner is relying on a combination of teachings from two distinct areas of the technology, specifically the UNIX (i.e., the AAPA) operating system environment and the WindowsNT (the "Custer" reference) operating system environment. Applicants respectfully assert that one having skill in the relevant art would not look to a WindowsNT-based system or reference when

addressing the scheduling of operations in a UNIX-based system. The two approaches are distinct and are not compatible since one (i.e., the UNIX approach) is a non-threaded, processor-based environment while the other (i.e., the WindowsNT approach) is a threaded environment. Not only would one having skill in the UNIX art not look to the WindowsNT art, but also the scheduling for one would simply not be applicable, and consequently not operable, for the other. Applicants direct the Examiner's attention to an article entitled "NT vs. UNIX: Is One Substantially Better?" from "WindowsNT Magazine" dated December 1998, which expressly states that "[o]ne notable difference is that NT boosts the priorities of dynamic threads...whereas UNIX depresses dynamic threads' priorities as the threads consume the CPU. Both OSs try to treat CPU-bound and I/O-intensive threads fairly with respect to other threads, but each OS goes about this task differently." Clearly, as evidenced by the foregoing "teaching away" from combination of the references, it is well recognized in the relevant art that the two operating systems are distinct and cannot logically be combined.

What is taught and claimed for the present invention is a scheduling method and system for a UNIX-based environment. As is explicitly set forth in the independent claims, at least one local scheduler prioritizes ***processes*** in accordance with a global prioritized schedule which is generated at the global scheduler means. None of the teachings from the cited AAPA art regarding UNIX-based systems provides any teaching or suggestion of such a local prioritizing of processes based on a global schedule. Moreover, it would not be appropriate to suggest that the AAPA-UNIX teachings be modified by the teachings of the Custer reference, since Custer is directed to a WindowsNT system. As noted above, the WindowsNT system is a threaded environment wherein one would have to prioritize and schedule threads within processes (which is the subject of Applicants other related work), but not prioritize the actual processes.

Applicants respectfully submit that the Examiner has repeatedly rejected the claimed invention based on the combining of teachings which simply should not be combined. It is well established under U.S. Patent law that there must be some suggestion or motivation for one having skill in the

relevant art to combine the references. Not only is there no such suggestion or motivation, but Applicants have submitted an article which expressly teaches away from such combination. Furthermore, it is well established that any combination of teachings which would result in an unworkable embodiment is patently non-obvious, and that a rejection based on such an attempted combination cannot be sustained. As such, Applicants respectfully request that the Examiner withdraw the rejections which are based on a combination of non-compatible UNIX and WindowsNT references.

In a further attempt (i.e., sixth amendment) to provide language which the Examiner will deem patentable, Applicants have submitted herein amendment language for the independent claims which explicitly recites that the invention is directed to a UNIX-based system wherein local processes are dynamically prioritized in accordance with a global schedule. Applicants also respectfully request that the Examiner consider the claims amendments and contact the undersigned attorney to schedule an interview to discuss the status of the application.

Applicants had previously requested that the Examiner directly address the arguments presented. In response, the Examiner has included a ***Response to Arguments*** located at paragraphs 7 and 8 in the Final Office Action. In addition to the above-provided arguments, Applicants would like to respond to the Examiner's statements found in the ***Response to Arguments*** section.

In paragraph 7a, the Examiner indicates asserts that the Zolnowsky queue is being correlated to the global prioritized schedule of the present invention. Applicants respectfully disagree. The Zolnowsky references teaches the use of queues. In a queue, an operation is held back from execution until it has been selected by a scheduler; at which time, only one operation is selected. In the invention as taught and claimed, however, all operations are already resident in and dispatched to the CPU for execution. Nothing is waiting in a queue. The global prioritized list provides a way for the local scheduler to favor execution of one resident operation over another. Clearly, the claim language of dynamically creating a global prioritized schedule for dispatched operations is not the same as statically queuing operations.

In paragraph 7b, the Examiner first avers that "the local prioritized schedule is updated-Not required by the limitations of Claim 11." Applicants respectfully disagree and direct the Examiner's attention to the last claim feature/limitation of Claim 11 in which the step of "dynamically prioritizing said local processes in accordance with said global prioritized schedule to allow simultaneous execution of tasks from said more than one application" is expressly recited.

Next, in paragraph 7b, the Examiner states that the claims' limitations of creating and communicating a global prioritized schedule are obviated by the Zolnowsky queue teachings, and quotes page 8 of Applicants' remarks from the previously-submitted amendment (paper 15) in which Applicants discuss the Zolnowsky queue. What the Applicants were pointing out is that under Zolnowsky's teachings, one operation goes to a processor from the global queue. Once the single global operation is provided to the processor, the processor compares the global operation to a local operation, and selects which one to do first. What is provided from the global entity (i.e., the global queue) is a single operation. The Zolnowsky processor does not

receive a global schedule of prioritized operations...it receives one operation. In contrast, under the present invention, all operations are already dispatched to the processor, the processor receives the global schedule of multiple prioritized operations, and the processor utilizes the global prioritized schedule to assign priorities to processes locally. Clearly, the comparison of one operation dispatched from a queue is not suggestive of providing a global prioritized schedule and utilizing same to assign local process priorities.

In paragraph 7c, the Examiner states that the Custer threads are associated with a process. Applicants do not disagree with that statement. What the Applicants disagree with is the Examiner's conclusion that since a WindowsNT system has threads within processes, such should be combined with the UNIX system teachings (i.e., the AAPA) to obviate the claim language. Applicants maintain the earlier-stated arguments that the references should not be combined, that relevant art teaches away from the combination of the two types of teachings, and that any such combination would not result in a workable system (and clearly would not result in the invention as claimed).

With regard to paragraph 7d, the Examiner has concluded that "local scheduling is not required to be accomplished by a software process distinct from the global scheduling algorithms." Applicants do not understand what the Examiner is saying in this sub-paragraph. In Claim 1, and those claims which depend therefrom, two distinct schedulers are recited. Similarly, in Claim 11 and those claims which depend therefrom, two distinct steps are recited. The Applicants' argument regarding nodes having a plurality of processes was not meant to recite a local and a global scheduling process at each node. What the Applicants meant was that each node is capable of scheduling and executing multiple operations with its plurality of processes.

In paragraph 7e, the Examiner concludes that the claims' limitation "does not require tasks from more than one application to execute on any one processor at any moment." Applicants respectfully assert that Applicants' statement that the invention allows the simultaneous execution of tasks from more than one application (Claim 1, lines 2-3 and 20-21, and Claim 11, line 19) is not the same as the Examiner's statement of "[requiring]...tasks from more than one application to execute on any one processor at

any moment". Applicants again refer the Examiner's attention to the previously-submitted arguments and request clarification of the Examiner's statements found in paragraph 7e.

Finally, with regard to paragraph 7f, Applicants again point out that the WindowsNT threaded approach with threads within processes is not relevant to or appropriately cited against the present invention.

As to the ***Response to Arguments*** section found in paragraph 8, Applicants submit the following remarks. As to sub-paragraph 8a, Applicants assert that a multiprocessor environment is not necessarily a multi-node environment. Each node can be a multiprocessor environment, and the system can include a plurality of such nodes. Applicants see no contradiction in what was stated in Applicants' previous remarks.

As to paragraph 8b, Applicants reiterate the arguments presented above to the effect that the claim limitation did not refer to separate processes for local and global scheduling at each node.

As to paragraph 8c, Applicants again assert that the use of two queues, one for affined and one for non-affined

processes, does not amount to the creation of a global schedule. What results when one utilizes the Boland queues is a system wherein a next-up affined process is compared to a next-up non-affined process and one is selected for execution (a la Zolnowsky). Such is not the same as, nor suggestive of, creating and communicating a global prioritized schedule and then prioritizing processes locally based on the global schedule.

As to paragraph 8d, Applicants first assert that the Examiner's conclusions from paragraph 8c and from paragraph 8d are seemingly inconsistent. Moreover, Applicants reiterate that the provision of a next-up operation from a queue is not the same as providing a global prioritized schedule for a plurality of tasks.

As to paragraph 8e, Applicants assert that the Claim 11 language of "dynamically prioritizing local processes" is a recitation of the limitation. Applicants further assert that re-ordering a queue and then providing a single next-up operation is not the same as providing a global prioritized schedule for use by the local scheduling entity.

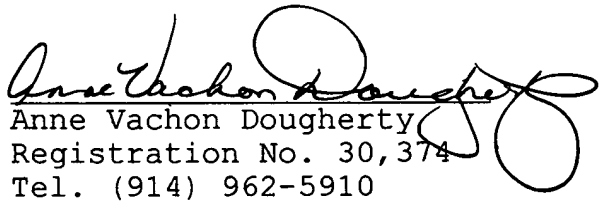
Finally, as to paragraph 8f, Applicants again note that the Examiner is inappropriately combining AAPA (i.e., UNIX)

and Custer (i.e., WindowsNT) teachings. Applicants rely on the earlier-recited arguments in support of the contention that the references should not be combined and that the combination would not yield the invention as claimed.

In light of the foregoing amendments and remarks, Applicants respectfully request entry of the amendments, reconsideration of the arguments, and allowance of the claims.

Respectfully submitted,

M.A. COHEN, ET AL

By: 
Anne Vachon Dougherty
Registration No. 30,374
Tel. (914) 962-5910